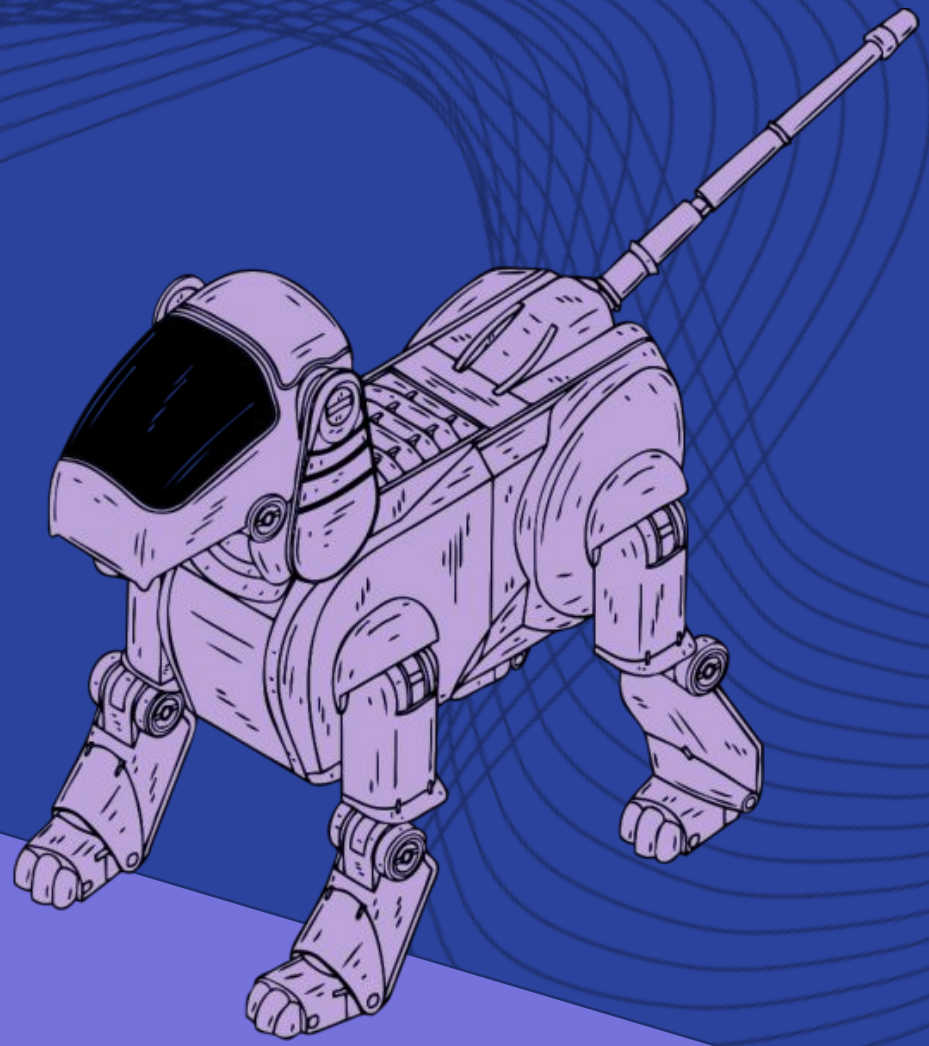
Spiral maker

Level 3 – Turtle Graphics in Python

Arts and Creativity



cair 4 YOUTH

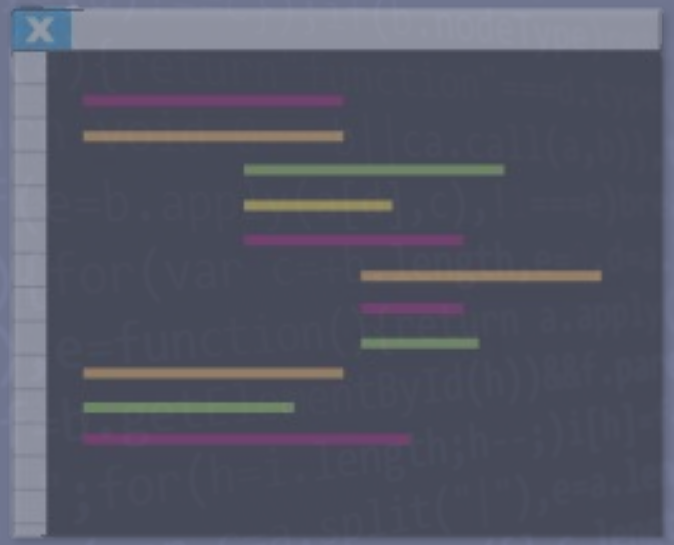
Introduction to Coding

What is Python?

- Python is a popular general-purpose programming language that can be used for a wide variety of applications.
- Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high-level dynamic data types, and classes.

Where to access Python?

- https://www.python.org/downloads/ - downloadable app for PCs (allows you to save files directly onto a computer)
- https://trinket.io/ - online version (allows you to create an account, much like scratch)



cair
4 YOUTH

Introduction

Spirals in modern art

Artists all over the world have been using the principles of geometric spirals for centuries. Nods to the artistic movement can be seen in the works of Van Gough's famous "Starry Night" oil painting, as well as modern artists like American, painter and printmaker Louise Bourgeois, whose oil-based woodcuts made on Japanese handmade paper are entitled "spirals".

cair
4 YOUTH

cair
4 YOUTH

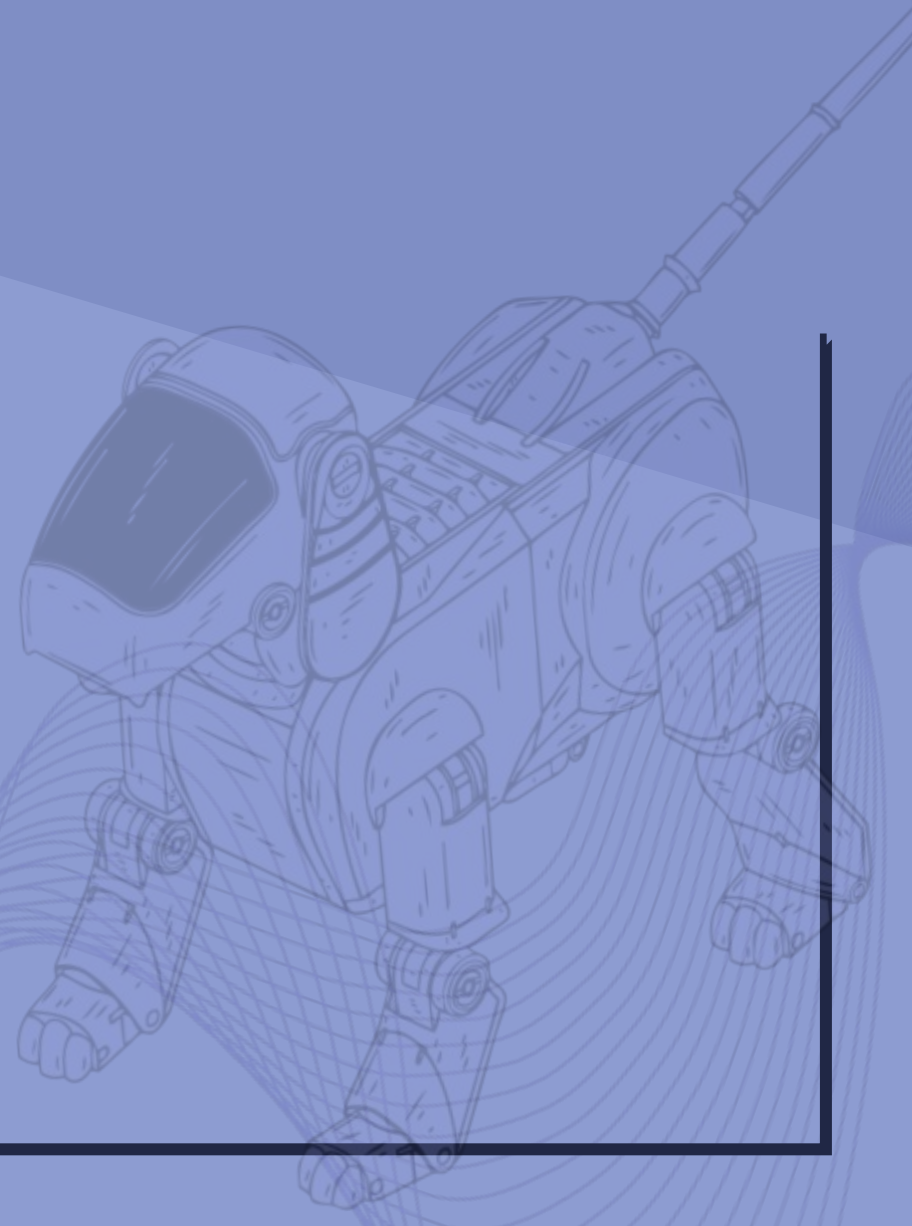
Task

Your task is to create a code in turtle graphics that enables users to create spirals (dependent on the parameters entered).

Process

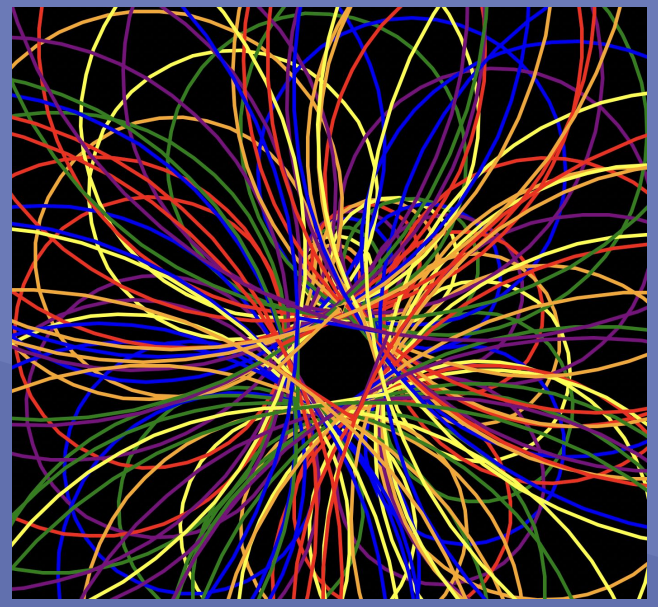
Your code should...

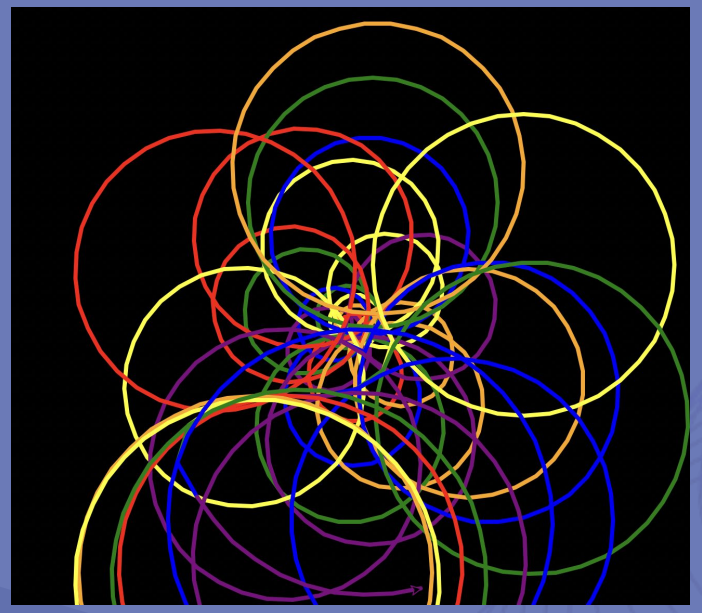
- Allow the user to enter parameters for change in the size of the spiral shape, the speed of the turtle, the pen thickness used to draw the spiral, and the angle at which the next shape is drawn compared to the last.
- Draw a spiral on the screen made up of the same shape, that rotates in different directions and changes colour.

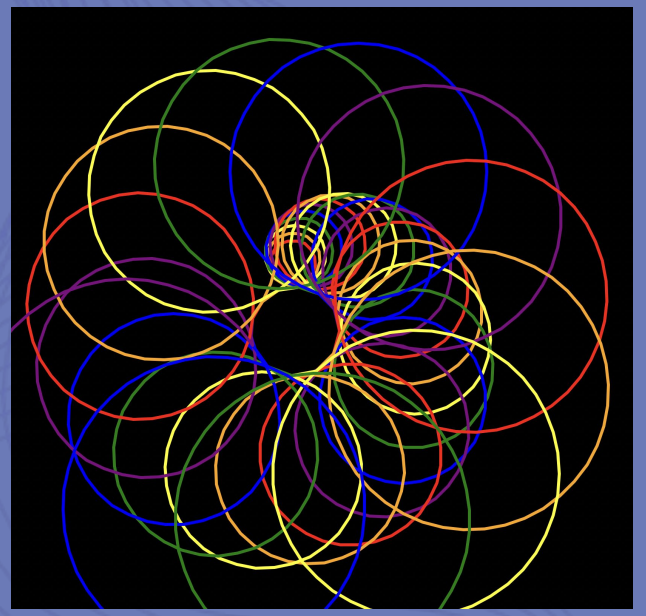


cair
4 YOUTH

What the spirals could look like







Depending on the parameters entered by the user and the time that the spiral is left for, depends on how the final spiral will look. Here are some examples.

Python libraries

Python libraries are a set of useful functions that eliminate the need for writing codes from scratch.

```
import turtle
import statistics
import random
```

They can be brought into the programme using the “import” keyword and can save valuable time when writing complex programmes.

One common example of a python library that we will be using in this code is the “turtle” library which enables us access to graphical functions in python.

Subroutines

Subroutines are sets of instructions designed to perform a frequently used operation within a programme.

```
1
2 def greeting():
3     print("Hello World!")
4     print("How are you today?")
5
6
7 greeting()
8
```

```
Hello World!
How are you?
```

Subroutines can store code and will only be run when ‘called’.

There are two main types of the subroutine: procedures and functions.

Procedures are not required to return a value, whereas functions must return a value.

Subroutines are great ways of writing more maintainable code and lead to more structured, organized and understandable programmes.

Parameters and arguments

Information can be passed into subroutines using parameters, which act as placeholders for the real values assigned when called.

The values put in the brackets when the subroutine is called are called arguments.

```
1
2 def addition(num1, num2):
3     result = num1 + num2
4     print(num1,"+",num2,"=",result)
5
6 num1 = input("Enter a number: ")
7 num2 = input("Enter a number: ")
8
9 addition(num1, num2)
```

```
Enter a number: 10
Enter a number: 9
10 + 9 = 19
```

Step 1

Importing libraries

```
1 import turtle
2 from itertools import cycle
3 colours=cycle(["red","orange","yellow","green","blue","purple"])
```

First, we will import the graphical ”turtle” library on line one of the code, enabling us to draw the spiral

The next library that will be imported is the “cycle” library which can be found in “itertools”. It is a simple process that will loop through a list in the order that it was written. On line 3, you can see that the list we will be using is called colours, and will store the different colours that we will use throughout the programme.

Simple turtle commands

The turtle understands simple commands such as left turns, right turns, move forwards and move backwards.

Simple Turtle commands:

- .forward() – moves by the amount specified in the brackets.
- .right() – moves the turtle clockwise by an angle specified in the brackets.
- .pencolor()- colour is spelled in the American way, but the colour is specified in the brackets.
- .circle()- is a built-in turtle command for drawing a circle where the radius is specified in the brackets.

```
turtle.forward(howmany)
turtle.right(angle)
turtle.pencolor("green")
turtle.circle(size)
```

Step 2

drawCircle() subroutine- declaring and calling

```
5 def drawCircle(size,angle,shift):
```

The next thing in the code will be the declaring of the drawCircle() subroutine. As can be seen in the brackets, the parameters that are passed in are the size of the circle, the angle to shift the circle by each time, and the direction that the circle is shifted in.

These parameters will be entered as numbers when the subroutine is called later in the program as can be seen below.

```
12 drawCircle(30,0,1)
```

Recursion

When a subroutine calls itself, this is called recursion. It’s another way of making a loop in your programme.

In most uses, the parameters of the function change each time the function is called. In our spiral programme, for example, the size, angle and position are changed every time the subroutine (or function) calls itself.

Step 3

Recursion and changing the circle each time

As you can see, the subroutine has been called again, however this time, the parameters are changed slightly each time.

This will mean that the spiral will move around the page differently each time depending on what is entered.

```
5 def drawCircle(size,angle,shift):
6     turtle.pencolor(next(colours))
7     turtle.circle(size)
8     turtle.right(angle)
9     turtle.forward(shift)
10    drawCircle(size + 5, angle + 1, shift + 1)
11
```

Different parameters mean different spirals can be drawn, have a go at experimenting by entering different values in line 10!

```
drawCircle(size + 10, angle + 4, shift + 2)
```

```
drawCircle(size + 9, angle + 13, shift + 56)
```

Step 4

Finalising the background colours, speed and pen size

```
12 turtle.bgcolor("black")
13 turtle.speed("fast")
14 turtle.pensize(4)
```

The background colour is set to be black so that the spiral really stands out, however, see what happens when you enter this line of code into your drawCircle() subroutine (before recursion).

```
turtle.bgcolor(next(colours))
```

The pen size can be changed, to increase the thickness of the pen.

Speed “fast” will enable you to see what is being drawn as it happens.

What the final code should look like...

Have a go at changing the values were highlighted to different numbers and see what your spiral looks like.

```
1  import turtle
2  from itertools import cycle
3  colours=cycle(["red","orange","yellow","green","blue","purple"])
4
5  def drawCircle(size,angle,shift):
6      turtle.pencolor(next(colours))
7      turtle.circle(size)
8      turtle.right(angle)
9      turtle.forward(shift)
10     drawCircle(size + 5, angle + 1, shift + 1)
11
12 turtle.bgcolor("black")
13 turtle.speed("fast")
14 turtle.pensize(4)
15
16 drawCircle(30,0,1)
17
```

Links to everyday life

Modern art

Research into different artists that use geometric and artistic spirals can lead to some interesting spirals being created by students.

Creativity

It is important to integrate creativity into coding. It makes programming more fun and leads to some amazing and innovative creations.

Spirals in nature

Spirals are also a natural occurrence such as the shape formed by the wind in a tornado (also known as a twister).

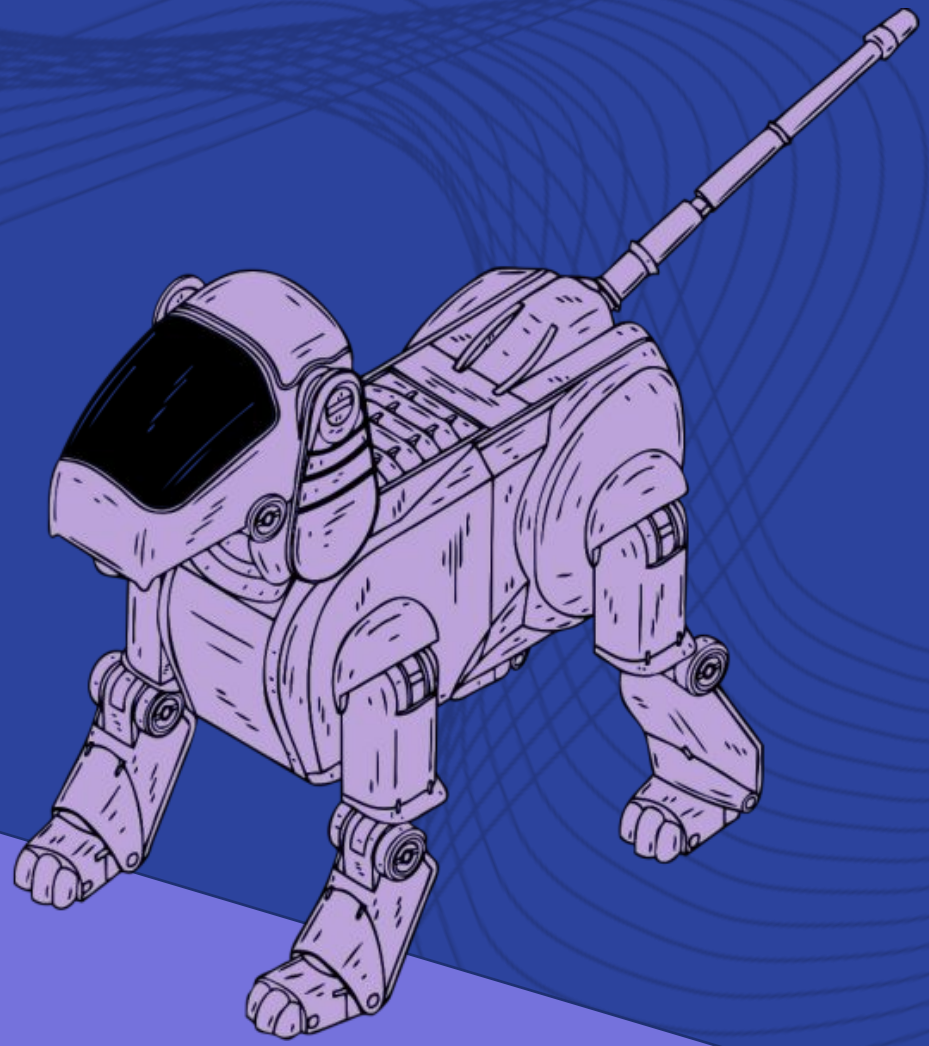
Conclusion

Learning outcomes

- ✔ You should be able to use subroutines with recursion as a method of iteration (repeating code).
- ✔ You should be able to use turtle graphics to create a circle, at varying positions on the screen.

Congratulations!

You have completed spiral creator



cair
4 YOUTH